

Knowledge-based systems in Bioinformatics, 1MB602

Lecture 10: Decision trees

Lecture overview

- Classification
- Decision trees
- Information theory
- Gain criterion
- Gain ratio
- Over fitting and pruning
- Application guidelines
- Bioinformatics examples

Classification

- The task of creating rules or structures (classification model) that will group individuals into predetermined classes
- Supervised approach
 - The algorithm has knowledge of the classes into which individuals fall when constructing these structures
- Example questions:
 - What features make an individual prone to sunburn?
 - What are the genetic differences between diseased individuals and normal individuals?
- Mutually exclusive classes
 - ‘Sick’ vs. ‘Healthy’
- Select those features which are most strongly associated with a particular classification for each sample
 - The fewer features used, the better the classification
- Interpretation of model important

Decision (Identification) trees

- Proven very successful in the classification domain
 1. Relatively undemanding in computational terms
 2. Provide clear, explicit reasoning of the decision making in the form of decision trees which can be converted to sets of rules
 3. They are accurate and robust in the face of noise
- DTs are trees of features that provide **tests** for classifying the samples in the data according to their most important features
- Basic premise:
 - Only a few features are required to classify all samples
 - Problem: identify this set of features
- Approach:
 - Test each feature iteratively to identify its potential for dividing the samples into classes

Example

- Umpires' decision to play a cricket match
 - Data on three factors thought to influence the decision

Weather	Light	Ground condition	Umpires' decision
Sunny	Good	Dry	Play
Overcast	Good	Dry	Play
Raining	Good	Dry	No play
Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Raining	Poor	Damp	No play
Overcast	Good	Damp	Play
Sunny	Poor	Dry	Play

- Task: determine the rules the umpires are explicitly or implicitly using

Decision tree algorithm

- Aim: split the data so that each subset of the data uniquely identifies a class in the data
- Algorithm summary:
 1. For each feature, compute the gain criterion
 2. Select the best feature and split the data according to the values in that feature
 3. If each of the subsets contains only one decision value, then stop. Otherwise reapply 1-3 on each of the subsets of data
 4. If the data is not completely classified but there are no more splits available then stop

Cricket game

- Need to divide the set of training examples into two smaller sets: 'Play' and 'No play'

- $Light = Good$ yields four examples:

Sunny	Good	Dry	Play
Overcast	Good	Dry	Play
Raining	Good	Dry	No play
Overcast	Good	Damp	Play

- $Light = Poor$ yields four examples:

Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Raining	Poor	Damp	No play
Sunny	Poor	Dry	Play

- What feature to use for splitting is determined using a measurement of its effectiveness

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Gain criterion

- Based on the amount of information that a test on the data conveys (information theory)
- The information contained within a test is related to the probability of selecting one training example from the training set T from a class C_j :

$$-\log_2 \left(\frac{\text{freq}(C_j, T)}{|T|} \right)$$

- Information (measured in bits):
- Expected information from the training set (k different classes):

$$in(T) = \sum_{j=1}^k \frac{\text{freq}(C_j, T)}{|T|} * \log_2 \left(\frac{\text{freq}(C_j, T)}{|T|} \right)$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Gain criterion

- Information yielded by a split x :

$$in_x(T) = -\sum_{i=1}^n \frac{|T_i|}{|T|} * in(T_i)$$

where n is the number of values for feature x and T_i is the subset of the training set given by the i th value of x

- The gain given by a particular test:

$$\text{gain}(X) = in(T) - in_x(T)$$

- The decision tree algorithm proceeds through each feature, computing the gain criterion, selects the best of these and then uses the same method for the remaining subsets

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Cricket game

Start with the hypothesis that no features are important and then check each feature in turn

Weather	Light	Ground condition	Umpires' decision
Sunny	Good	Dry	Play
Overcast	Good	Dry	Play
Raining	Good	Dry	No play
Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Raining	Poor	Damp	No play
Overcast	Good	Damp	Play
Sunny	Poor	Dry	Play

$$in(T) = -4/8 \cdot \log_2(4/8) - 4/8 \cdot \log_2(4/8) = 1$$

(Play) (No play)

$$in_{\text{weather}}(T) = 2/8 \cdot (-2/2 \cdot \log_2(2/2)) + 4/8 \cdot (-2/4 \cdot \log_2(2/4) - 2/4 \cdot \log_2(2/4)) + 2/8 \cdot (-2/2 \cdot \log_2(2/2))$$

(Sunny) (Overcast) (Raining)

$$= 0.5 \text{ bits}$$

$$\text{Gain} = 1 - 0.5 = 0.5$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Cricket game

Weather	Light	Ground condition	Umpires' decision
Sunny	Good	Dry	Play
Overcast	Good	Dry	Play
Raining	Good	Dry	No play
Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Raining	Poor	Damp	No play
Overcast	Good	Damp	Play
Sunny	Poor	Dry	Play

$$in_{\text{light}}(T) = 4/8 \cdot (-3/4 \cdot \log_2(3/4) - 1/4 \cdot \log_2(1/4)) + 4/8 \cdot (-1/4 \cdot \log_2(1/4) - 3/4 \cdot \log_2(3/4))$$

(good) (poor)

$$= 0.811 \text{ bits}$$

$$\text{Gain} = 1 - 0.811 = 0.189$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Cricket game

Weather	Light	Ground condition	Umpires' decision
Sunny	Good	Dry	Play
Overcast	Good	Dry	Play
Raining	Good	Dry	No play
Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Raining	Poor	Damp	No play
Overcast	Good	Damp	Play
Sunny	Poor	Dry	Play

$$in_{\text{light}}(T) = 5/8 \cdot (-3/5 \cdot \log_2(3/5) - 2/5 \cdot \log_2(2/5)) + 3/8 \cdot (-1/3 \cdot \log_2(1/3) - 2/3 \cdot \log_2(2/3))$$

(dry) (damp)

$$= 0.951 \text{ bits}$$

$$\text{Gain} = 1 - 0.951 = 0.049$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

First split: weather feature

- Splits in three subsets: Sunny, Overcast, and Raining
- Overcast subset (S) needs a new split!

Overcast	Good	Dry	Play
Overcast	Poor	Dry	No play
Overcast	Poor	Damp	No play
Overcast	Good	Damp	Play

$$in(S) = -2/4 \cdot \log_2(2/4) - 2/4 \cdot \log_2(2/4) = 1$$

$$in_{right}(S) = 2/4 \cdot (-2/2 \cdot \log_2(0/2) - 0/2 \cdot \log_2(0/2)) \quad (good) \\ + 2/4 \cdot (-0/2 \cdot \log_2(0/2) - 2/2 \cdot \log_2(2/2)) \quad (poor) \\ = 0 \text{ bits}$$

$$Gain = 1 - 0 = 1$$

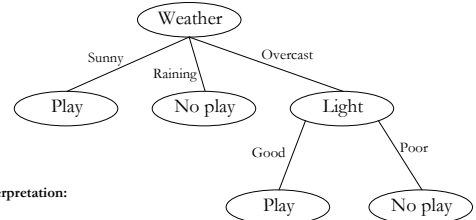
$$in_{ground}(S) = 2/4 \cdot (-1/2 \cdot \log_2(1/2) - 1/2 \cdot \log_2(1/2)) \quad (dry) \\ + 2/4 \cdot (-1/2 \cdot \log_2(1/2) - 1/2 \cdot \log_2(1/2)) \quad (damp) \\ = 1 \text{ bits}$$

$$Gain = 1 - 1 = 0$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Cricket game

Final decision tree:



Interpretation:

- IF weather = sunny THEN play
- IF weather = raining THEN no play
- IF weather = overcast AND light = good THEN play
- IF weather = overcast AND light = poor THEN no play

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Continuous data

- Only considered discrete case
- Real world examples are often continuous
- Do the same as in the discrete cases but
 - Swap the = operator with other comparison operators (<, <=, >, >=)

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Gain ratio

- Gain criterion biased towards tests which have many subsets (weather feature)
 - Tests that result in many subsets are not necessarily those that will yield the most useful information
- Idea: take into account the size of the subsets created by the test
 - Divide the gain by the information contained by the number of subsets in the split (split information measure)

$$splitin(X) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \cdot \log_2 \left(\frac{|T_i|}{|T|} \right)$$

$$gainratio(X) = \frac{gain(X)}{splitin(X)}$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Overfitting and pruning

- Every algorithm involved with classification runs the risk of overfitting the data
 - The alg. learns the errors (noise) in the data as well as the underlying structure of the processes that created the data
 - Occurs because the alg. tries to reduce the classification error
- To identify this phenomenon:
 - Split data into training data (≈75%) and test data (≈25%)
 - Build tree on the training data and test the model on the test data
 - A tree X is overfitted if there exists a tree Y that do better on the unseen test set, but worse on the training set
- Prune complex branches of the tree
 - Results in less accurate trees for the training data
 - Post-pruning: Use some estimate of the expected error of:
 - The current subtree
 - A leaf that could replacing the subtree
 - Pre-pruning: Stop increasing the size of a subtree when the information gained is below some threshold

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Other disadvantages of DT

- The algorithm shown here generates only one tree based on information gain
 - This is a greedy strategy
 - More accurate trees with worse start splits may exist!
- Would benefit from some search strategy
 - A split could be evaluated in terms of its current ability to classify the data AND the accuracy of the splits later on in the algorithm run
 - This will unavoidably increase the time complexity of the algorithm

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Application guidelines

- DTs are useful when there are a large number of records in the data
- Restricted to classification problems where the class of the training examples are known (supervised learning)
- In bioinformatics, the number of examples is often small in comparison to the number of attributes
 - Not feasible to split the data into separate large training and test sets
 - Use a **cross-validation** scheme where the alg. is run separately on different training sets and test sets
 - Split the data into a number of folds and repeatedly train on n-1 of the data and test on the last fold
 - Repeat for all the other n-1 folds (n-fold cross-validation)
 - At each run measure the error
 - Average the errors as a measure of accuracy

CS
http://www.lch.lu.se

Multiple decision trees

- Li et al., 2003
 - Use a committee of trees to determine the clinical diagnosis of an individual
 - Avoids the deterministic features of the DT algorithm (top ranked attribute make the split)
 1. Build a tree from the best feature
 2. Build another tree from the second best feature and so on up to a stopping point
 3. Convert the trees into rules and add them to a knowledge base
 - During classification, use the coverage statistics (number of individual records covered by the rule) as a measure of the generality of each rule
 - The coverage for each rule that fire is summed for each class and the class with highest sum is predicted

B
IATICS
http://www.lch.lu.se

Consensus method for secondary structure prediction

- Secondary structure:
 - Determines how groups of amino acids form sub-structures
 - Provides vital information as to the tertiary structure and therefore the function of the protein
- Selbig et al., 1999
 - Used DTs to combine predictions of other methods (DSSP, DEFINE) – **meta-classifier**
 - IF Method1 = Helix AND Method2 = Helix THEN CONSENSUS = Helix
 - Prediction performance at worst the same as the best prediction method

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lch.lu.se

References

- E. Keedwell, A. Narayanan, Intelligent bioinformatics: the application of artificial intelligence techniques to bioinformatics problems. Chichester : John Wiley, cop. 2005
- S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice-Hall, Upper Saddle River, New Jersey, 1995

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lch.lu.se